Supplementary File: Transfer Learning from Synthetic to Real-Noise Denoising with Adaptive Instance Normalization

1. Transfer Learning from AWGN

We present the results of transfer-learned denoiser where AINDNet is pre-trained with AWGN and adapted to real noise (RN). For the precise comparison, we report performance of three denoisers in Table 1 according to training sets and learning methods:

- AINDNet(AWGN): AINDNet is trained with AWGN images.
- AINDNet(AWGN)+TF$_1$: AINDNet(AWGN) is transfer learned with a single real noisy image.
- AINDNet(AWGN)+TF: AINDNet(AWGN) is transfer learned with full real noisy images (320 images).

It can be seen that proposed transfer learning scheme significantly improves the performance of synthetic noise (SN) denoisers including AWGN denoiser when the input is limited.

Table 1: Average PSNR of the denoised images on the SIDD validation set. $_1$ denotes that the number of real training noisy image is one.

Method	PSNR
RIDNet [2]	38.71
AINDNet(S)	35.21
AINDNet(AWGN)	26.25
AINDNet(R)	38.81
AINDNet(AWGN)+TF	38.82
AINDNet+TF	38.90
AINDNet(R)$_1$	30.36
AINDNet(AWGN)+TF$_1$	31.76
AINDNet+TF$_1$	36.19

2. More Noise Level Estimation Results

We evaluate the accuracy of the proposed noise level estimator, where the input images are simultaneously corrupted with more diverse signal-dependent noise levels σ_s and signal-independent noise levels σ_c. As presented in Table 2, the proposed noise level estimator achieves better accuracy with lower standard deviations of the errors in most cases. Furthermore, the proposed noise level estimator predicts quite accurate estimates when the images are corrupted with high σ_s and σ_c.

Table 2: Average MAE and error STD for the images from Kodak24 where the inputs are corrupted by heteroscedastic Gaussian including in-camera pipeline.

Method	FCN [6]		Ours	
(σ_s, σ_c)	MAE	STD	MAE	STD
(0.04, 0.00)	**0.009**	**0.007**	0.022	0.014
(0.04, 0.02)	0.029	**0.007**	**0.015**	0.011
(0.04, 0.04)	0.050	**0.006**	**0.009**	0.009
(0.04, 0.06)	0.070	**0.007**	**0.016**	0.009
(0.08, 0.00)	**0.018**	**0.013**	0.022	0.014
(0.08, 0.02)	0.039	0.013	**0.014**	**0.012**
(0.08, 0.04)	0.059	0.014	**0.012**	**0.011**
(0.08, 0.06)	0.076	0.013	**0.020**	**0.010**
(0.12, 0.00)	0.029	0.020	**0.020**	**0.014**
(0.12, 0.02)	0.052	0.021	**0.015**	**0.014**
(0.12, 0.04)	0.071	0.020	**0.017**	**0.014**
(0.12, 0.06)	0.087	0.020	**0.030**	**0.014**
(0.16, 0.00)	0.039	0.027	**0.021**	**0.018**
(0.16, 0.02)	0.065	0.028	**0.020**	**0.019**
(0.16, 0.04)	0.076	0.027	**0.021**	**0.019**
(0.16, 0.06)	0.098	0.028	**0.040**	**0.021**
Average	0.054	0.017	**0.020**	**0.014**
# params	29.5 K		29.7 K	

3. More Visualized Results

We present more visualized comparisons on three test sets: SIDD, RNI15 and DND. We compare proposed methods with conventional methods *i.e.* KSVD [1], BM3D [5], MLP [3], TNRD [4], DnCNN [8], TWSC [7], CBDNet [6], and RIDNet [2].

We visualize the results of the proposed methods and previous methods in Fig. 1 and 2 where the noisy images are achieved from SIDD. It can be seen in Fig. 1 that AINDNet(S) and AINDNet+TF infer edge preserved results with clearer characters than conventional methods. Although the results of AINDNet(S) look good with more vivid color and edge preservation, it infers some color distortion. On the other hand, AINDNet+TF retains color information and also processes severe color-changing regions neatly, where other methods produce mosaic-like patterns. The visual comparisons on RNI15 are also shown in Fig. 3. AINDNet(S) and AINDNet+TF infer edge preserved results, so denoised characters and hair are more visually pleasing than in other methods. Lastly, we compare proposed methods with other methods on DND where quantitative results and processed images can be achieved from DND sites. As presented in Fig. 4 and 5, AINDNet(S) and AINDNet+TF get the best quantitative results and also achieve well-denoised images. Specifically, AINDNet(S) and AINDNet+TF remove noises while preserving the edges of the engraving and the textures in Fig. 4. From these visual comparisons, including the ones in the main manuscript, we believe that AINDNet(S) can infer edge-preserved results by training a lot of training set, and this learned knowledge is effectively transferred to AINDNet+TF.

4. Ablation Study

We demonstrate the effectiveness of noise level estimator for training with $\mathcal{S}$. We present performance of noise level estimators combined with reconstruction network in Table 3 with different objective function. Remember that $L_{ms\text{-}asymm}$ can generate smoothed outputs, so L_{TV} is excluded when using $L_{ms\text{-}asymm}$. We find that state-of-the-art training scheme (FCN + L_{asymm} + L_{TV}) infers inferior performance than proposed training scheme (Ours + $L_{ms\text{-}asymm}$). Moreover, the proposed training scheme also surpasses internal variation (Ours + L_1 + L_{TV}).

Table 3: Investigation of noise level estimator and estimation loss when denoisers are trained with SN data. The quantitative results (in average PSNR (dB)) are reported on DND test dataset and SIDD validation dataset.

Method	DND	SIDD
FCN + L_{asymm} + L_{TV}	39.51	34.90
Ours + L_1 + L_{TV}	39.45	35.08
Ours + $L_{ms\text{-}asymm}$	**39.53**	**35.19**

We further investigate the relation between update parameters and performance in the transfer learning phase. For the precise comparison, we compare three variants by freezing each update parameter in Table 4:

- Ours-AIN: AIN module is not updated in transfer learning stage.
- Ours-Estimator: Noise level estimator is not updated in transfer learning stage.
- Ours-LastConv: Last convolution is not updated in transfer learning stage.

It can be seen that proposed updating the noise level estimator, and last convolution contribute 0.1 - 0.2 dB performance gain respectively. Fixing AIN module parameter presents even worse performance than the SN denoiser.

Table 4: Investigation of update parameters when denoisers are transfer-learned with RN data. The quantitative results (in average PSNR (dB)) are reported on SIDD validation dataset.

Method	PSNR
Ours-AIN	34.60
Ours-Estimator	38.71
Ours-LastConv	38.75
AINDNet(S)	35.21
AINDNet+TF	38.90

References

[1] Michal Aharon, Michael Elad, and Alfred Bruckstein. K-svd: An algorithm for designing overcomplete dictionaries for sparse representation. *IEEE Transactions on signal processing*, 54(11):4311–4322, 2006. 1

[2] Saeed Anwar and Nick Barnes. Real image denoising with feature attention. In *The IEEE International Conference on Computer Vision (ICCV)*, October 2019. 1

[3] Harold C Burger, Christian J Schuler, and Stefan Harmeling. Image denoising: Can plain neural networks compete with bm3d? In *2012 IEEE conference on computer vision and pattern recognition*, 2012. 1

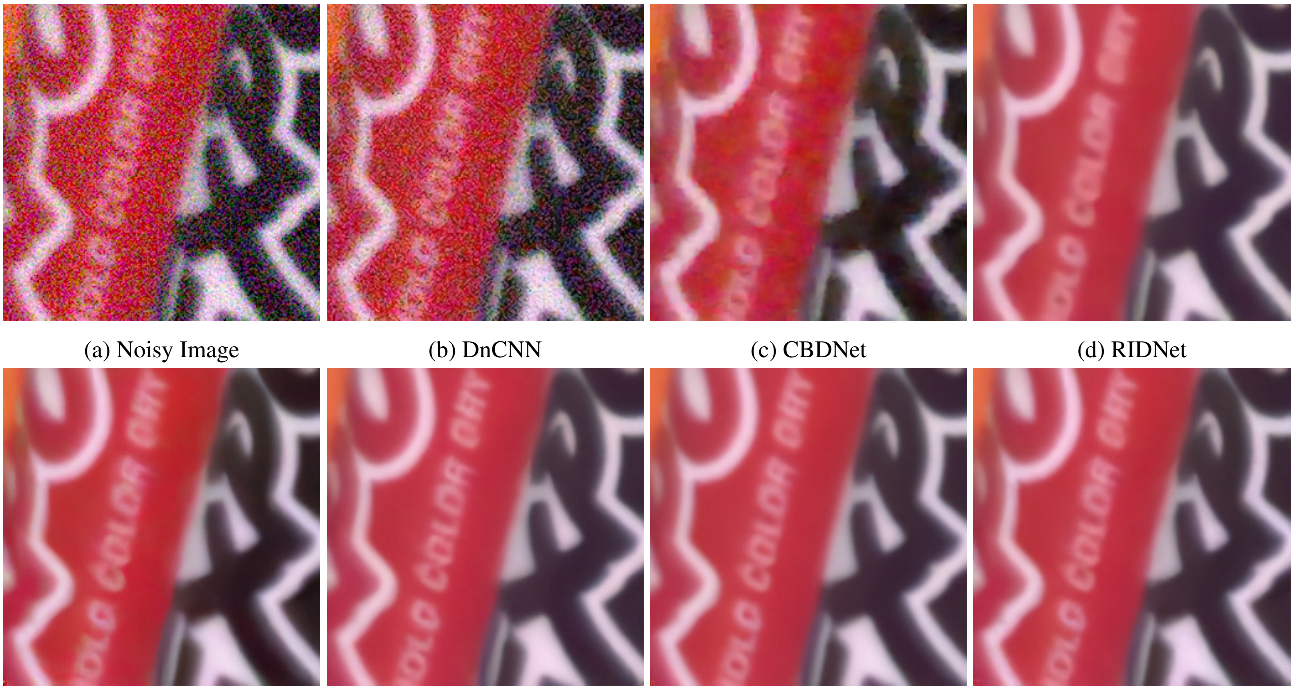
[4] Yunjin Chen and Thomas Pock. Trainable nonlinear reaction diffusion: A flexible framework for fast and effective image restoration. *IEEE transactions on pattern analysis and machine intelligence*, 39(6):1256–1272, 2016. 1

[5] Kostadin Dabov, Alessandro Foi, Vladimir Katkovnik, and Karen Egiazarian. Color image denoising via sparse 3d collaborative filtering with grouping constraint in luminance-chrominance space. In *2007 IEEE International Conference on Image Processing*, 2007. 1

[6] Shi Guo, Zifei Yan, Kai Zhang, Wangmeng Zuo, and Lei Zhang. Toward convolutional blind denoising of real photographs. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019. 1

[7] Jun Xu, Lei Zhang, and David Zhang. A trilateral weighted sparse coding scheme for real-world image denoising. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018. 1

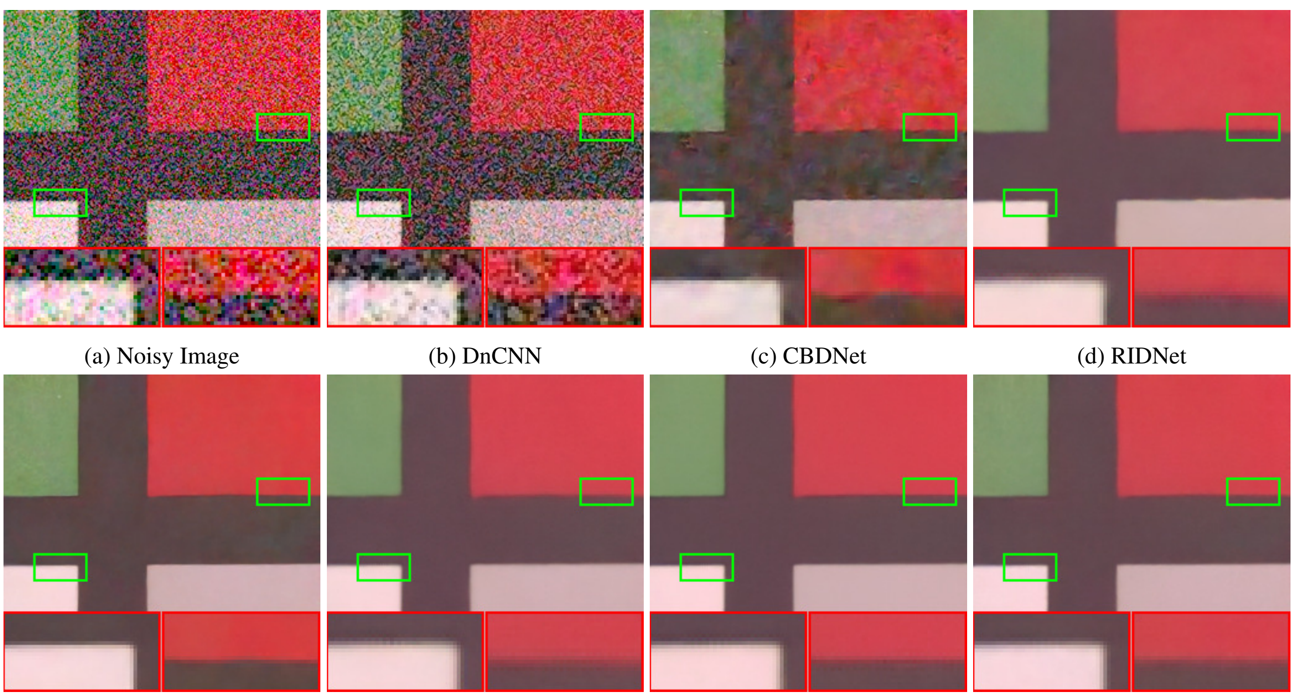
[8] Kai Zhang, Wangmeng Zuo, Yunjin Chen, Deyu Meng, and Lei Zhang. Beyond a gaussian denoiser: Residual learning of deep cnn for image denoising. *IEEE Transactions on Image Processing*, 26(7):3142–3155, 2017. 1



(a) Noisy Image (b) DnCNN (c) CBDNet (d) RIDNet

(e) AINDNet(S) (f) AINDNet(R) (g) AINDNet+RT (h) AINDNet+TF

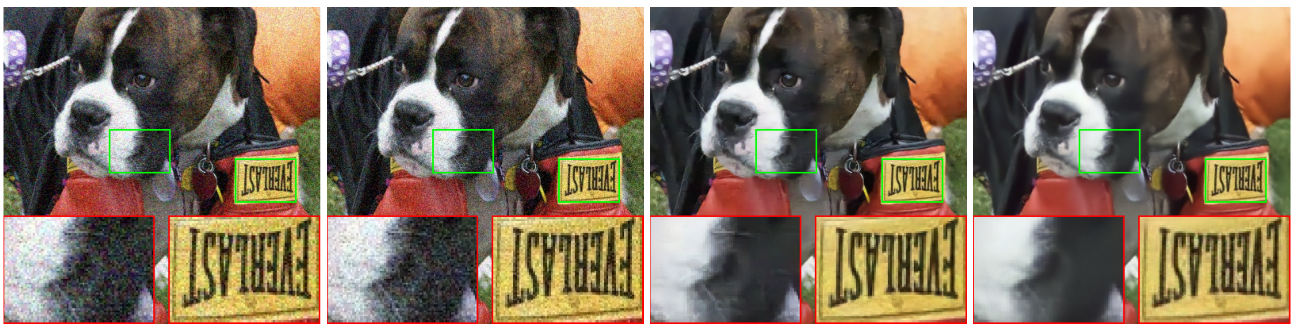
Figure 1: A real noisy image from SIDD, and the comparison of the results.



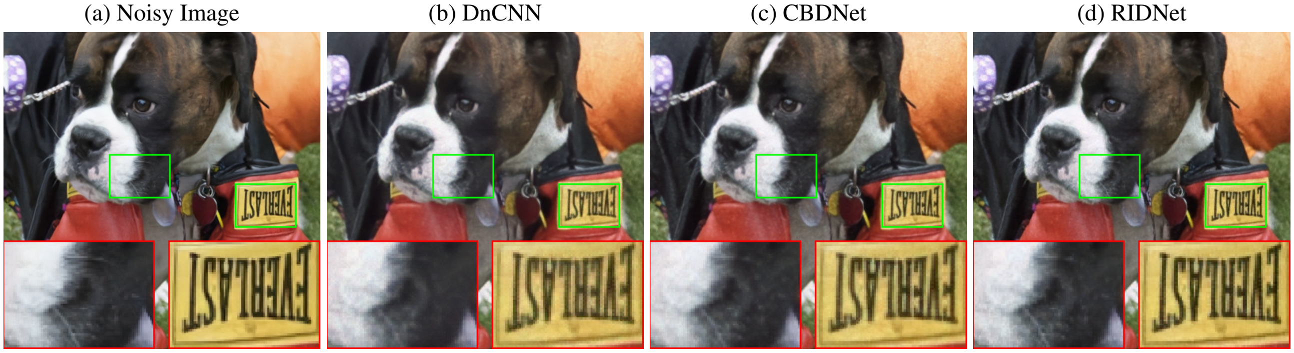
(a) Noisy Image (b) DnCNN (c) CBDNet (d) RIDNet

(e) AINDNet(S) (f) AINDNet(R) (g) AINDNet+RT (h) AINDNet+TF

Figure 2: A real noisy image from SIDD, and the comparison of the results.

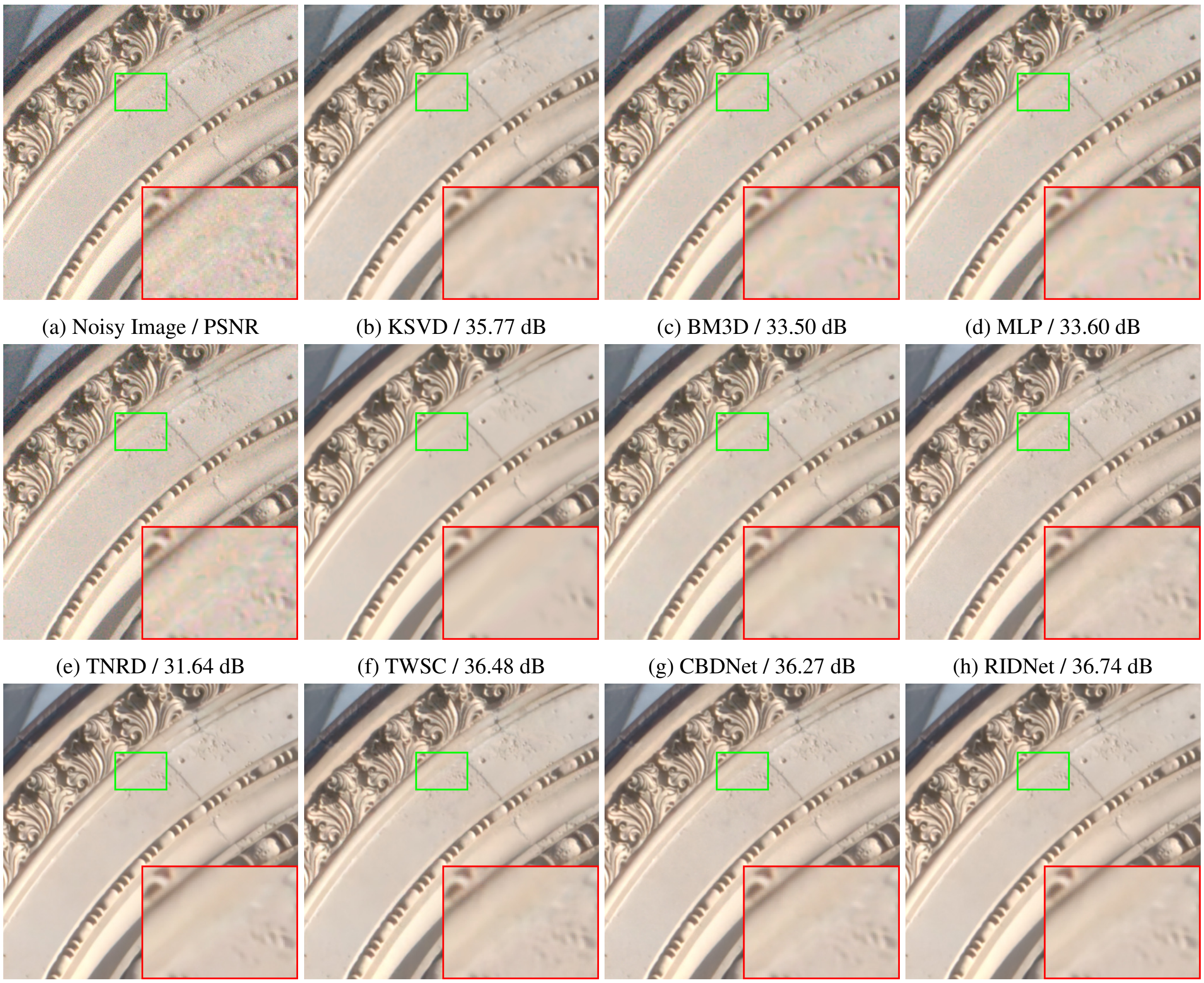


(a) Noisy Image (b) DnCNN (c) CBDNet (d) RIDNet



(e) AINDNet(S) (f) AINDNet(R) (g) AINDNet+RT (h) AINDNet+TF

Figure 3: A real noisy image from RNI15, and the comparison of the results.

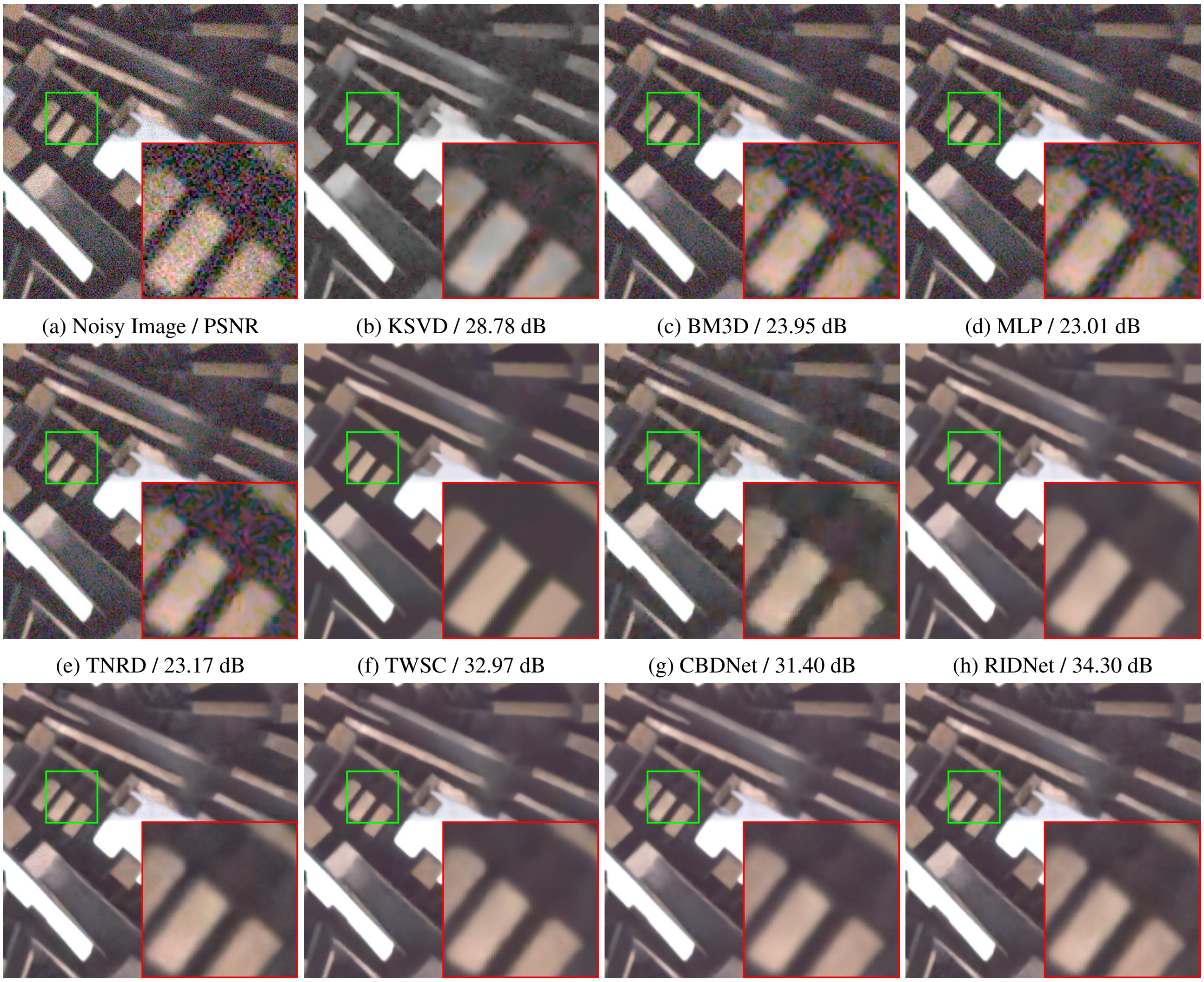


(a) Noisy Image / PSNR (b) KSVD / 35.77 dB (c) BM3D / 33.50 dB (d) MLP / 33.60 dB

(e) TNRD / 31.64 dB (f) TWSC / 36.48 dB (g) CBDNet / 36.27 dB (h) RIDNet / 36.74 dB

(i) AINDNet(S) / 37.15 dB (j) AINDNet(R) / 37.60 dB (k) AINDNet+RT / 37.49 dB (l) **AINDNet+TF / 38.09 dB**

Figure 4: A real noisy image from DND, and the comparison of the results.



(a) Noisy Image / PSNR | (b) KSVD / 28.78 dB | (c) BM3D / 23.95 dB | (d) MLP / 23.01 dB

(e) TNRD / 23.17 dB | (f) TWSC / 32.97 dB | (g) CBDNet / 31.40 dB | (h) RIDNet / 34.30 dB

(i) **AINDNet(S) / 34.63 dB** | (j) AINDNet(R) / 33.90 dB | (k) AINDNet+RT / 32.62 dB | (l) AINDNet+TF / 33.39 dB

Figure 5: A real noisy image from DND, and the comparison of the results.